



Funciones

Transcripción

[00:00] Otro camino para resolver nuestro problema de no repetir líneas de código con la misma introducción, en este caso los saltos de línea es usando funciones. El concepto de funciones dentro de programación no es más que un código que recibe instrucciones que puede o no recibir algunos parámetros de entrada, o sea, inputs de algunas informaciones y nos devuelve un resultado.

[00:32] Básicamente eso es lo que hace una función dentro de programación. Son muy utilizadas y ayudan bastante en el desarrollo del código.

[00:43] Son superimportantes porque me ayudan a ser muy eficientes a la hora de trabajar con códigos y de organizar todo mi proyecto, de organizar todo mi trabajo que estoy haciendo, porque puedo reutilizar una función las veces que yo quiera, las veces que yo necesite, incluso le puedo pasar diferentes parámetros para que me retorne resultados diferentes.

[01:13] Lo vamos a ver ahora en la práctica para que se entienda mejor ese concepto de funciones. Aquí nosotros ya estamos usando funciones que vienen por default, que ya fueron creadas en JavaScript, por ejemplo document write, habíamos usado alert, ¿recuerdan que lanzaba un pop-up, un mensaje de alerta en la pantalla? Entonces estas funciones ya vienen por defecto en JavaScript.

[01:50] Importante en las funciones, que siempre inician y terminan con un paréntesis, y algunas de ellas reciben un parámetro, el caso de alert, recibe un

parámetro texto que va a ser el mensaje que me va a lanzar en la pantalla. En el caso de document write es una función que me devuelve ese texto en la pantalla de nuestro navegador, en este caso, que es el browser que estamos usando.

[02:20] Entonces las funciones tienen esa característica, de iniciar con un paréntesis, enviamos un parámetro, cerramos el paréntesis, y el punto y coma que es una buena práctica de programación en JavaScript. Entonces vamos a crear una función que sea salto de línea, que ejecute lo mismo, básicamente, que haga lo mismo que estamos haciendo aquí con la función por default, por defecto, que viene en JavaScript.

[02:53] ¿Por qué? Porque a la hora de otra persona, una tercera persona, mire nuestro código, esto de aquí, hasta es entendible, pero si la persona no sabe lo que significa document write, si esa persona no es de programación, ella no va a saber que lo que estamos haciendo es llamar a una función que va a mostrar en tela lo que estamos mandando en este parámetro.

[03:20] Entonces es mucho más fácil si en mi código yo tengo una función que se llame vamos a colocar saltar línea, una convención de programación, otra más de las tantas que hay es que en las funciones siempre ocupemos verbos, porque una función normalmente la mayoría de las veces representa una acción. En este caso va a ser yo quiero saltar línea. Vamos a crear una función que sea saltar línea.

[03:52] Imagínense que mi función no recibe nada como parámetro, y en este caso vamos a aprender una función que no reciba nada como parámetro, para eliminar esta parte de aquí. Entonces cuando ejecute saltar línea, va a hacer ese salto de línea que estamos pidiendo en función de la cantidad de veces que vamos a definir aquí nuestra función.

[04:19] ¿Cómo se crea una función? Tenemos que usar la palabra function, que es del inglés, función en inglés. Siempre se acostumbra usar las funciones al inicio

del código, en este caso ni bien comenzamos nuestro código de JavaScript vamos a crear nuestra función.

[04:42] Aquí coloco los dos paréntesis. De momento nuestra función no va a tener parámetros, más adelante lo vamos a aprender cómo es con parámetros, y aquí vamos a escribir nuestra función que va a ser saltar línea. Luego, ¿cuál es otro elemento de las funciones? Son las llaves.

[05:07] Las llaves significan el inicio y el fin de nuestra función. Todo lo que coloquemos aquí adentro van a ser las instrucciones que esa función va a realizar, y en nuestro caso voy a colocar que nuestra función va a ser document write BR. Punto y coma. Otra convención, otra buena práctica es la indentación, indentar nuestro código, igual dentro de las funciones.

[05:51] Entonces así yo voy a saber que todo lo que está dentro e indentado son las instrucciones de esta función. Me permite visualmente saber que todo lo que voy a colocar aquí dentro de las llaves y con una indentación son las instrucciones de esta función saltar línea.

[06:14] Entonces ya creé mi función. Solo repasando todas las características de ella necesito un nombre. Ese nombre explícito usando un verbo, aquí está saltar línea. Entonces es un verbo, tiene que abrir y cerrar paréntesis, en este caso particular no tenemos parámetros todavía, lo vamos a ver más adelante. Necesito una llave que abre y que cierra, que es el inicio y el fin de mi función.

[06:43] Y aquí mi instrucción es que en este momento va a ser un salto de línea. Entonces, esta es la definición de mi función. ¿Cómo es que yo llamo a esa función? Con el nombre, abriendo y cerrando paréntesis, saltar línea, abro y cierro paréntesis. Y aquí voy a sustituir, voy a reemplazar ese saltar línea, abro, cierro, punto y coma. Voy a guardar, "Ctrl + S", vengo aquí, actualizo. Y subimos un pequeño error.

[07:30] ¿Qué pasó? Aquí. Nos olvidamos una parte importante, que eran los corchetes. Con eso, solucionado, nuestro código debería funcionar. Perfecto. Entonces, tenemos un salto de línea, porque definimos solo un salto de línea, y es lo que estamos viendo aquí. Un salto de línea. Entonces cuando el código llega aquí, primeramente el código crea la función y la deja ahí cargada por si la vamos a usar, es lo que hace en este caso el compilador.

[08:13] Cuando él comienza a ejecutar todas las instrucciones de nuestro código, para aquí en la 19 y ve, hay una llamada a la función. Entonces, ejecuta a lo que se está llamando. Llega, sigue, sigue, 25, hay una llamada a la función. Ejecuta a lo que se está llamando. Y aquí, dentro de mi función ahora, si nuestro jefe nos dice: "Quiero dos". Perfecto.

[08:35] Vengo en un lugar solo que yo sé dónde está, porque va a estar en la función saltar línea, coloco un salto, imagínense que nos dice: "Quiero tres saltos". Perfecto. Coloco los saltos de línea, voy a salvar, vengo a mi programa y actualizar, y tengo los saltos de línea actualizados.

[09:02] Reforzando la importancia aquí de los paréntesis, si yo los dejo en blanco, no va a pasar absolutamente nada porque no estoy llamando una función. No me está reconociendo nada, ni siquiera ese código, y este Pedro está viniendo a la misma línea porque no ejecuté ningún salto. Entonces, importante el uso de paréntesis, importante también nuevamente el uso de nomenclaturas que sean explícitas.

[09:40] Usen esa nueva práctica. Si yo hubiera colocado X, X y X. ¿Qué significa eso? ¿Va a funcionar? Va a funcionar. Hasta lo voy a guardar para mostrarles y funciona perfectamente. Pero de aquí a dos meses tengo que hacer mantenimiento, tengo que hacer una lectura y un entendimiento de lo que significaba eso para poder darle mantenimiento a ese código.

[10:09] Entonces, es mejor acostumbrarnos desde un inicio a usar nombres que sean de fácil entendimiento y que representen en este caso la acción que estoy pidiendo a mi función. Bueno, más adelante lo que vamos a hacer es ver funciones con parámetros que reciben un input, en este caso no recibía, para continuar repasando y aprendiendo estos conceptos muy importantes en lógica de programación.